Efficient Performance Modeling of Analog Integrated Circuits via Kernel Density Based Sparse Regression

Chenlei Fang[1], Qicheng Huang[1], Fan Yang[1,*], Xuan Zeng[1,*], Dian Zhou[1,2], and Xin Li[1,3]
[1]State Key Lab of ASIC & System, Microelectronics Department, Fudan University, Shanghai, P. R. China
[2]Electrical Engineering Department, University of Texas at Dallas, Richardson, TX, USA
[3]Electrical & Computer Engineering Department, Carnegie Mellon University, Pittsburgh, PA, USA

ABSTRACT

With the aggressive scaling of integrated circuit technology, analog performance modeling is facing enormous challenges due to high-dimensional variation space and expensive transistor-level simulation. In this paper, we propose a kernel density based sparse regression algorithm (KDSR) to accurately fit analog performance models where the modeling error is not simply Gaussian due to strong nonlinearity. The key idea of KDSR is to approximate the non-Gaussian likelihood function by using non-parametric kernel density estimation. Furthermore, we adopt Laplace distribution as our prior knowledge to enforce a sparse pattern for model coefficients. The unknown model coefficients are finally determined by using an EM type algorithm for maximum-a-posteriori (MAP) estimation. Our proposed method can be viewed as an iterative and weighted sparse regression algorithm that aims to reduce the estimation bias for model coefficients due to outliers. Our experimental results demonstrate that our proposed KDSR method can achieve superior accuracy over the conventional sparse regression method.

1. INTRODUCTION

As device size scales down into nanoscale region, variations introduced by manufacturing process start to affect analog circuit performance significantly. Such variations are closely related to yield deterioration and, hence, need careful modeling and analysis to guarantee circuit quality [1]-[2]. Over the past two decades, performance modeling techniques have been developed to address this variability issue [3]-[6]. These methods attempt to approximate the circuit performance of interest (e.g., power of an amplifier) as an analytical function of process variations (e.g., ΔV_{th}, ΔW, etc.). Once the performance models are created, they can be used to guide circuit optimization [7]-[11], find worst-case corners [12] and estimate parametric yield [13].

Although performance modeling has been successfully applied to many practical applications, several recent trends of IC design pose a number of new technical challenges in this area. Because of the new design methodologies adopted by analog designers, today's analog circuits and systems are increasingly complex with enormous number of devices, thereby leading to a two-fold consequence. First, the variation space becomes high-dimensional due to the large number of independent random variables that must be used to capture device mismatches. Second, the computational cost of transistor-level simulation significantly increases and is prohibitively expensive for complex analog circuits and systems.

Several advanced modeling techniques have recently been proposed in the literature to address this complexity issue. For instance, elastic net regularized learning [5] and Orthogonal Matching Pursuit (OMP) [6] exploit the underlying sparse pattern to solve a large number of model coefficients from few training samples without over-fitting. Bayesian Model Fusion (BMF) takes advantage of the data collected at an earlier stage to further improve modeling efficiency [15].

However, most existing performance modeling methods assume that the modeling error simply follows a Gaussian distribution so that an exponential likelihood function can be derived and used to solve the unknown model coefficients. Such a simple assumption, however, does not necessarily hold in practice. For instance, if process variations are extremely large and the transistors of an operational amplifier are no longer biased in the saturation region, the amplifier gain can be extremely small. Due to this reason, when we sample the variation space to generate training samples, a small number of training samples may become outliers and, therefore, make the simple Gaussian assumption invalid. If these outliners are not appropriately handled, they can strongly bias the modeling coefficients and prevent us from generating accurate performance models.

In this paper, we propose a novel modeling method, referred to as *Kernel Density based Sparse Regression* (KDSR), to accurately estimate performance models with consideration of outliers (i.e., non-Gaussian error distribution). Our key idea is to approximate the non-Gaussian likelihood function by using non-parametric kernel density estimation based upon the recent advances in statistics theory [14]. Furthermore, we adopt Laplace distribution as our prior knowledge to enforce a sparse pattern for model coefficients. Finally, a novel Expectation-Maximization (EM-type) algorithm is used to solve the unknown model coefficients by maximum-a-posteriori (MAP) estimation. Our proposed method can be viewed as an iterative and weighted sparse regression algorithm that is particularly developed to address the outlier issue for performance modeling. Our experimental results demonstrate that the proposed KDSR method substantially improves the modeling accuracy over the conventional sparse regression approach.

The remainder of this paper is organized as follows. In Section 2, we briefly review the background of performance modeling. The KDSR method is presented in Section 3. The efficacy of our proposed method is demonstrated by several experimental examples in Section 4. Finally, Section 5 concludes the paper.

2. BACKGROUND

Performance model is a broadly used approach to approximate the circuit performances with respect to the process variations. In this section, we briefly review the formulation of performance

* Corresponding authors: {yangfan, xzeng}@fudan.edu.cn.

DAC '16, June 05-09, 2016, Austin, TX, USA
 ISBN 978-1-4503-4236-0/16/06…$15.00
DOI: http://dx.doi.org/10.1145/2897937.2898013

modeling, along with existing model fitting methods and their limitations.

2.1 Performance Modeling

A performance model aims to approximate the circuit performance as an analytical function of variations with M basis functions:

$$y(\mathbf{x}) \approx \sum_{m=1}^{M} \beta_m \cdot g_m(\mathbf{x}) . \tag{1}$$

where y is the model function of the performance, $\mathbf{x}$ denotes the random variables representing device-level variations, $\{\beta_m; m=1 \ldots M\}$ mean the model coefficients, and $\{g_m(\mathbf{x}); m=1 \ldots M\}$ are the M basis functions of the random variables (e.g., linear or quadratic polynomials).

A performance model is specifically defined by the model coefficients. Once the model coefficients are determined, the model can be further used to estimate parametric yield, find worst-case corners, optimize circuit design, etc.

2.2 Least-Squares Fitting

In order to find model coefficients defined in (1), least-squares (LS) fitting method collects simulation samples and constructs the following linear equation:

$$\mathbf{G} \cdot \boldsymbol{\beta} = \mathbf{y} , \tag{2}$$

where

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_M \end{bmatrix}^T , \tag{3}$$

$$\mathbf{y} = \begin{bmatrix} y^{(1)} & y^{(2)} & \cdots & y^{(N)} \end{bmatrix}^T , \tag{4}$$

$$\mathbf{G} = \begin{bmatrix} g_1(\mathbf{x}^{(1)}) & g_2(\mathbf{x}^{(1)}) & \cdots & g_M(\mathbf{x}^{(1)}) \\ g_1(\mathbf{x}^{(2)}) & g_2(\mathbf{x}^{(2)}) & \cdots & g_M(\mathbf{x}^{(2)}) \\ \vdots & \vdots & & \vdots \\ g_1(\mathbf{x}^{(N)}) & g_2(\mathbf{x}^{(N)}) & \cdots & g_M(\mathbf{x}^{(N)}) \end{bmatrix} . \tag{5}$$

In (3)-(5), $\{\mathbf{x}^{(n)}, y^{(n)}\}$ represent the n-th sampling point. There are N samples collected by circuit simulations in total. LS fitting requires the number of sampling points to be greater than the number of equations, because it aims to solve an over-determined equation.

We can also view the process of finding the model coefficients as a maximum likelihood estimate (MLE). LS fitting method models the regression error ε as a zero-mean Gaussian distribution with variance σ_0:

$$y(\mathbf{x}) = \sum_{m=1}^{M} \beta_m \cdot g_m(\mathbf{x}) + \varepsilon , \tag{6}$$

$$pdf(\varepsilon) = \sqrt{\frac{1}{2\pi\sigma_0^2}} \cdot \exp\left(-\frac{\varepsilon^2}{2\sigma_0^2}\right) . \tag{7}$$

The log-likelihood of observing the N samples $\{\mathbf{x}^{(n)}, y^{(n)}, n=1, 2, \ldots N\}$ can thus be expressed as:

$$\text{const} - \left(\frac{1}{2\sigma_0^2}\right) \sum_{n=1}^{N} \left(y^{(n)} - \sum_{m=1}^{M} \beta_m \cdot g_m\left(x^{(n)}\right) \right)^2 . \tag{8}$$

where

$$\text{const} = N \log\left(\sqrt{\frac{1}{2\pi\sigma_0^2}}\right) .$$

The idea of LS fitting is maximizing the log-likelihood (8), which is equivalent to the following problem:

$$\min_{\boldsymbol{\beta}} \sum_{n=1}^{N} \left(y^{(n)} - \sum_{m=1}^{M} \beta_m \cdot g_m\left(\mathbf{x}^{(n)}\right) \right)^2 . \tag{9}$$

If the modeling error does follow Gaussian distribution, LS regression gives exactly the same results as MLE. However, if the error distribution is not Gaussian, MLE would achieve higher accuracy compared with LS method.

For instance, assume that the "real" error follows the distribution demonstrated in Figure 1, which is far from Gaussian distribution. Instead, it has a heavy tail.

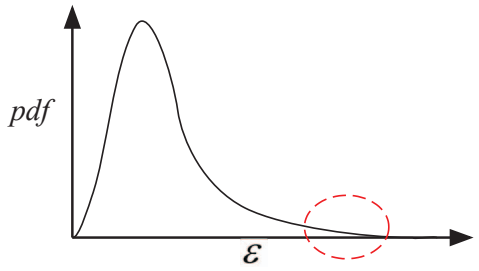


Figure 1. Error distribution with heavy tail

The sampling points around the tail (inside the red circle) have a larger deviation from zero. Because LS fitting tries to minimize the squares of fitting errors at each sampling point, these points around the tail will significantly impact the model coefficients. The fitted modeling error will be "shifted" to the tail from the real distribution. In other words, LS regression suffers from loss of accuracy in such scenario.

2.3 Sparse Regression and Bayesian Model Fusion

Another limitation related with least-squares fitting is that it requires the number of samples (i.e., N) to be larger than the number of coefficients (i.e., M). Due to the high simulation cost and the increasingly high dimension, this prerequisite becomes challenging.

Several advanced modeling approaches have been proposed to reduce the required sample size, aiming to take advantage of extra prior knowledge of model coefficients.

The first piece of prior knowledge is the sparsity. In real applications, only a few of the basis functions are actually required to approximate the performance model, which means many of the model coefficients are nearly zero. SR proposed in [5] exploits the underlying sparsity and solves the following convex optimization problem:

$$\min_{\boldsymbol{\beta}} \sum_{n=1}^{N} \left(y^{(n)} - \sum_{m=1}^{M} \beta_m \cdot g_m\left(\mathbf{x}^{(n)}\right) \right)^2 + \lambda_1 \|\boldsymbol{\beta}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|_2^2 , \tag{10}$$

where λ_1 and λ_2 are parameters controlling the regulation term.

OMP, on the other hand, solves the following L_1-norm regularization problem [6]:

$$\min_{\boldsymbol{\beta}} \sum_{n=1}^{N} \left(y^{(n)} - \sum_{m=1}^{M} \beta_m \cdot g_m\left(\mathbf{x}^{(n)}\right) \right)^2 \quad \text{subject to } \|\boldsymbol{\beta}\|_1 \le \lambda . \tag{11}$$

Another piece of prior information comes from the early-stage of circuit design, such as pre-layout simulation data. BMF [15] encodes the similarity of early-stage and late-stage data and gives the maximum-a-posteriori estimate:

$$\min_{\boldsymbol{\beta}} \sum_{n=1}^{N}\left(y^{(n)}-\sum_{m=1}^{M}\beta_m\cdot g_m\left(\mathbf{x}^{(n)}\right)\right)^2+\lambda\cdot\boldsymbol{\beta}^T\cdot\mathbf{D}\cdot\boldsymbol{\beta}\,, \tag{12}$$

where λ denotes the parameter controlling the regulation, and $\mathbf{D}$ means the covariance matrix learned from prior knowledge.

All the previous methods effectively reduce the sampling points needed for model fitting. However, the first terms in (10)-(12) are all the sum of error squares at every sampling point, which means that they suffer from the same issue with LS fitting.

3. PROPOSED APPROACH

In this section, we develop our proposed kernel density based sparse regression method. It is derived from a recently proposed kernel density based regression estimate (KDRE) in statistical field [14]. We further apply Laplace prior distribution to exploit the sparsity of model coefficients.

3.1 Kernel Density Estimation

We assume $f(t)$ to be the probability density of ε in (6). If $f(t)$ is known, MLE gives the estimation of model coefficients $\boldsymbol{\beta}$ by maximizing the log-likelihood:

$$\max_{\boldsymbol{\beta}} \sum_{n=1}^{N}\log f\left(y^{(n)}-\sum_{m=1}^{M}\beta_m\cdot g_m\left(\mathbf{x}^{(n)}\right)\right). \tag{13}$$

However, $f(t)$ is hardly available in practice, so (13) cannot be applied directly. Instead, the kernel density estimate finds a way to approximate the "real" error distribution based some initial parameter estimate.

Let $\boldsymbol{\beta}^{(0)}$ be an initial estimate of model coefficients (e.g. by LS fitting), and the residuals are given by:

$$\varepsilon_n^{(0)}=y^{(n)}-\sum_{m=1}^{M}\beta_m^{(0)}\cdot g_m\left(\mathbf{x}^{(n)}\right), \tag{14}$$

Kernel density based method estimates the error distribution $f(t)$ as a combination of kernel densities:

$$\tilde{f}(t)=\frac{1}{N}\sum_{n=1}^{N}K_h\left(t-\varepsilon_n^{(0)}\right), \tag{15}$$

where $K_h(t) = h^{-1}K(t/h)$. $K(\bullet)$ is called a kernel function. It is a non-negative function integrating to one with zero mean. The tuning parameter $h>0$ is called bandwidth. In this paper, we use the Gaussian kernel:

$$K(x)=\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right). \tag{16}$$

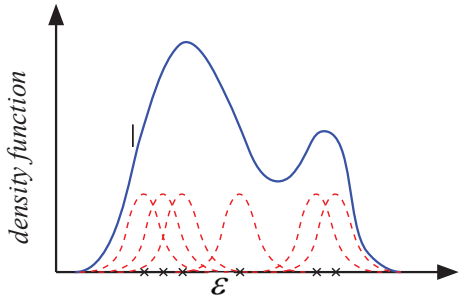


Figure 2. Error estimated with Gaussian Kernel

Figure 2 shows an example of kernel density estimation of error distribution. Markers on x-axis demonstrate initial residuals $\{\varepsilon_n^{(0)}, n=1,2,\ldots,N\}$. Red curves show N kernel densities $\{K_h(t-\varepsilon_n^{(0)}), n=1,2,\ldots,N\}$, each centering at an residual value. The blue curve shows linear combination of those kernel densities, corresponding to (15). As shown in Figure 2, kernel density estimate is a non-parametric way to estimate the probability density function with any shape.

Substitute $f(t)$ in (13) with (15), the kernel density based linear regression is given as:

$$\max_{\boldsymbol{\beta}} Q(\boldsymbol{\beta})\,, \tag{17}$$

where $Q(\boldsymbol{\beta})$ is the estimated log-likelihood function

$$\begin{aligned} Q(\boldsymbol{\beta}) &= \log pdf\left(\mathbf{x},\mathbf{y}\,|\,\boldsymbol{\beta}\right) \\ &= \sum_{n=1}^{N}\log\tilde{f}\left(y^{(n)}-\sum_{m=1}^{M}\beta_m\cdot g_m\left(\mathbf{x}^{(n)}\right)\right) \\ &= \sum_{n=1}^{N}\log\left[\frac{1}{N}\sum_{j\neq n}K_h\left(y^{(n)}-\sum_{m=1}^{M}\beta_m\cdot g_m\left(\mathbf{x}^{(n)}\right)-\varepsilon_j^{(0)}\right)\right] \end{aligned}. \tag{18}$$

We use leave-one-out kernel density to remove the estimation bias [14].

It has been proved in [14] that KDRE is asymptotically as efficient as the oracle MLE method that assumes we have already known the "real" error distribution.

3.2 Prior Definition

The KDRE method proposed in [14] generally handles the problem when error distribution is not Gaussian. However, the initial estimate $\boldsymbol{\beta}^{(0)}$ comes from LS estimate and the computation process still requires the number of sample points to be greater than the variable dimension, which is impractical.

To address this problem, we encode prior knowledge of performance modeling to improve the modeling efficiency. As described in Section 2, performance model coefficients are usually sparse. We can use Laplace prior distribution as our prior knowledge of the model coefficients:

$$pdf\left(\boldsymbol{\beta}\right)\propto\exp\left(-\kappa\cdot\left\|\boldsymbol{\beta}\right\|_1\right), \tag{19}$$

where κ is a hyper-parameter. The Laplace distribution decays when the L_1-norm of parameter increases, thus imposes sparsity on the model coefficients.

According to Baye's theorem, the posterior distribution of model coefficients is proportional to the product of prior distribution and likelihood function [18]:

$$pdf\left(\boldsymbol{\beta}\,|\,\mathbf{x},\mathbf{y}\right)\propto pdf\left(\boldsymbol{\beta}\right)\cdot pdf\left(\mathbf{x},\mathbf{y}\,|\,\boldsymbol{\beta}\right). \tag{20}$$

MAP method estimates the model coefficients by finding the value that maximizes the posterior distribution given by (20). Combining (18)-(20), MAP estimation can be equivalently written as the following optimization problem:

$$\max_{\boldsymbol{\beta}} \sum_{n=1}^{N}\log\left[\frac{1}{N}\sum_{j\neq n}K_h\left(y^{(n)}-\sum_{m=1}^{M}\beta_m\cdot g_m\left(\mathbf{x}^{(n)}\right)-\varepsilon_j^{(0)}\right)\right]-\kappa\left\|\boldsymbol{\beta}\right\|_1. \tag{21}$$

As for the initial parameter estimate $\boldsymbol{\beta}^{(0)}$, we can use the traditional sparse regression method with L_1-norm (SR-L1) [16]-[17]. SR-L1 uses the likelihood function (7) and the Laplace prior distribution(19), yielding the following optimization problem:

$$\max_{\boldsymbol{\beta}^{(0)}} \quad -\sum_{n=1}^{N}\left(y^{(n)}-\sum_{m=1}^{M}\beta_m^{(0)}\cdot g_m\left(\mathbf{x}^{(n)}\right)\right)^2-\lambda\left\|\boldsymbol{\beta}^{(0)}\right\|_1, \tag{22}$$

where λ is another hyper-parameter controlling the effect of L_1-norm item. Eq. (22) is convex and can be efficiently solved to find the global optimum.

Given the initial parameter estimate $\boldsymbol{\beta}^{(0)}$, we obtain the initial residuals by (14). We can also determine the rule of thumb bandwidth [14]:

$$h=1.06N^{-1/5}\sigma^{(0)}\,, \tag{23}$$

where $\sigma^{(0)}$ is the standard deviation of the initial residuals.

3.3 Maximum-a-posteriori Estimation

Since the optimization problem described in (21) has a mixture form and is not necessarily convex, we cannot directly find its optimum with convex optimization algorithms. We propose a novel Expectation-Maximization (EM) algorithm to calculate the optimization problem. Given the initial residuals estimation $\boldsymbol{\varepsilon}^{(0)}$ from (14), we then update the estimate by the following algorithm:

Algorithm 1: EM algorithm

Input: samples from simulation $\{(\mathbf{x}^{(i)}, y^{(i)}), i=1,2,\ldots,N\}$, initial residuals $\{\varepsilon_n^{(0)}, n=1,2,\ldots,N\}$, hyper-parameter κ, bandwidth h.
Output: Model coefficients $\boldsymbol{\beta}$

1: **for** $k=1 \rightarrow k_{max}$ **do**
2: E-step: Calculate the classification probabilities:

$$p_{ij}^{(k+1)} = \frac{K_h\left(y^{(i)} - \sum_{m=1}^{M} \beta_m^{(k)} \cdot g_m\left(\mathbf{x}^{(i)}\right) - \varepsilon_j^{(0)}\right)}{\sum_{l \neq i} K_h\left(y^{(i)} - \sum_{m=1}^{M} \beta_m^{(k)} \cdot g_m\left(\mathbf{x}^{(i)}\right) - \varepsilon_l^{(0)}\right)} \propto K_h\left(y^{(i)} - \sum_{m=1}^{M} \beta_m^{(k)} \cdot g_m\left(\mathbf{x}^{(i)}\right) - \varepsilon_j^{(0)}\right), \quad j \neq i \tag{24}$$

3: M-step: Update $\boldsymbol{\beta}^{(k+1)}$, solve

$$\max_{\boldsymbol{\beta}^{(k+1)}} \sum_{i=1}^{N} \sum_{j \neq i} \left\{ p_{ij}^{(k+1)} \log K_h\left(y^{(i)} - \sum_{m=1}^{M} \beta_m^{(k+1)} \cdot g_m\left(\mathbf{x}^{(i)}\right) - \varepsilon_j^{(0)}\right)\right\} - \kappa \left\|\boldsymbol{\beta}^{(k+1)}\right\|_1 \tag{25}$$

which can be written as:

$$\min_{\boldsymbol{\beta}^{(k+1)}} \sum_{i=1}^{N} \sum_{j \neq i} \left\{ p_{ij}^{(k+1)} \left(y^{(i)} - \sum_{m=1}^{M} \beta_m^{(k+1)} \cdot g_m\left(\mathbf{x}^{(i)}\right) - \varepsilon_j^{(0)}\right)^2 \right\} + \kappa \left\|\boldsymbol{\beta}^{(k+1)}\right\|_1 \tag{26}$$

4: **If** $\|\boldsymbol{\beta}^{(k+1)} - \boldsymbol{\beta}^{(k)}\|_1 < \delta$ (δ is a pre-defined small value) **then**
5: break
6: **end if**
7: **end for**

It should be noted that (26) is convex and can be solved efficiently by the convex optimization algorithms [19].

Studying (26), we can view the proposed method as a weighted version of (22). Eq. (26) attempts to minimize the weighted square difference between the new residual $y^{(i)} - \boldsymbol{\beta}^T \mathbf{x}^{(i)}$ and the initial residual $\varepsilon_j^{(0)}$ for all $1 \leq i \neq j \leq N$. The weights are obtained by (24). If the j-th sampling point $\{(\mathbf{x}^{(j)}, y^{(j)})\}$ is an outlier, $\varepsilon_j^{(0)}$ is quite large. For sampling points $\{(\mathbf{x}^{(i)}, y^{(i)}), i \neq j\}$, p_{ij} is small since we use Gaussian kernel. For the sampling points different from the j-th sampling point $\{(\mathbf{x}^{(j)}, y^{(j)})\}$, the Gaussian kernel decays rapidly. The effect of $\varepsilon_j^{(0)}$ on determining $\boldsymbol{\beta}^{(k+1)}$ will thus be small because the weights p_{ij} are small for the outlier $\{(\mathbf{x}^{(j)}, y^{(j)})\}$. By using the EM algorithm, KDSR can be considered as an iterative and weighted sparse regression, which is able to detect outliers.

3.4 Hyper-parameter Estimation

In (21), hyper-parameter κ controls the effect of sparse regulation. If κ is very large, the optimization process prefers sparse solutions. Therefore, fewer effective basis functions are chosen, creating a simple model but leaving large modeling error. Such scenario is called under-fitting. On the other hand, if κ is small, (21) is approximately the same as (17)-(18), and the estimation result only depends on the sampling points. However, if the sample size is not large enough, the obtained model may fit the sample points very well but fails to generalize to new samples, which causes over-fitting. Hyper-parameter λ affects (22) in a similar way.

Due to the reasons explained above, we must properly decide the hyper-parameter κ and λ to avoid either over-fitting or under-fitting. We adopt the idea of L-fold cross-validation in statistical field [18] to address this problem. The entire sample set is partitioned into L groups. Algorithm 1 runs L times with each candidate value of κ or λ. In each run, L-1 groups are used as training group to give the estimation of model coefficients by (26). The remaining group is considered as the testing group to estimate the modeling error's likelihood with kernel density estimate (15).

For each run, training groups and the testing group should not overlap, thus over-fitting can be easily detected. Each run produces a log-likelihood value. The modeling log-likelihood is thus given by the sum of all L log-likelihood values $\{L_l, l=1,2,\ldots,L\}$.

After performing cross-validation with each candidate hyper-parameter, the hyper-parameter producing the maximum log-likelihood on testing group is chosen as the final hyper-parameter.

3.5 Summary

Algorithm 2 summarizes the major steps of our proposed KDSR method.

Algorithm 2: Kernel Density Based Sparse Regression

Input: samples from simulation $\{(\mathbf{x}^{(i)}, y^{(i)}), i=1,2,\ldots,N\}$
Output: Model coefficients $\boldsymbol{\beta}$

1: Use cross-validation to calculate the hyper-parameter λ in (22).
2: Calculate the initial estimation of model coefficients $\boldsymbol{\beta}^{(0)}$ by solving the convex optimization problem in (22).
3: Compute the initial residuals $\{\varepsilon_n^{(0)}, n=1,2,\ldots,N\}$ and bandwidth h by (14) and (23).
4: Apply cross-validation to determine the hyper-parameter κ.
5: Use Algorithm 1 to get the kernel density based estimation of model coefficients $\boldsymbol{\beta}$.

4. NUMERICAL EXAMPLES

In this section, two circuit examples are used to verify the efficiency of our proposed KDSR method. We build the performance models of the circuits with two different methods: (i) the sparse regression method based on L_1-norm regularization (SR-L1) (ii) our proposed kernel density based sparse regression method (KDSR). The traditional SR-L1 method is demonstrated in (22). In the experiments, we use linear functions of the random variables as the basis functions $\{g_m(\mathbf{x}); m=1\ldots M\}$ in (1).

All experimental samples in this section are generated by independent Monte Carlo simulations with random samplings. We partition the data set into two non-overlapping parts, one as the training set and the other as the testing set. The training set is used for model fitting. The testing set estimates the modeling likelihood. The same process is performed 10 times, with different choices of training set and testing set to average randomness. All experiments are performed on a server with 2.5GHz dual-core CPU and 16GB memory.

4.1 Operational Amplifier

A two-stage operational amplifier designed in a commercial 45nm process is used in this example. Figure 3 shows a simplified schematic of this circuit. There are 581 independent random

variables to represent inter-die variations and random mismatches.

We aim to model gain of the circuit as a linear function of these random variables. In order to test the accuracy, we generate 5000 simulation samples. The histogram of gain is plotted in Figure 4(a). It shows that the amplifier fails to work at some sample points, and the corresponding gain values are extremely small, resulting in a tail of the gain distribution.

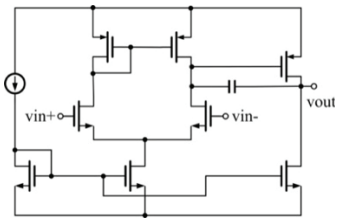
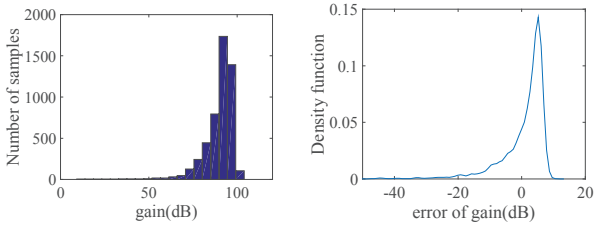


Figure 3. Simplified schematic of operational amplifier



(a) (b)

Figure 4. (a)The histogram of simulation data of gain (b) probability density function of "golden" error

For comparison, kernel density based linear regression estimate (KDRE) [14] is applied to the whole data set to obtain the "golden" model coefficients $\boldsymbol{\beta}_{real}$ and error distribution:

$$\varepsilon_{real}^{n} = y^{(n)} - \sum_{m=1}^{M} \beta_{real,m} \cdot g_m\left(\mathbf{x}^{(n)}\right) \tag{27}$$

where n=1,2,…,N_{total}. N_{total} is the size of the whole data set.

Given the "real" error samples, kernel density estimate can be used to approximate the probability density function. The "golden" error distribution is shown in Figure 4(b). It shows that the error distribution is far away from Gaussian distribution.

In this example, we use 3000 samples as the testing group. Two methods are applied for gain modeling: SR-L1 method and KDSR method. After obtaining the corresponding model coefficients $\boldsymbol{\beta}_{SR\text{-}L1}$ and $\boldsymbol{\beta}_{KDSR}$, we first calculate the fitting error of the testing set:

$$\varepsilon_{SR-L1}^{(i)} = y^{(i)} - \sum_{m=1}^{M} \beta_{SR-L1,m} \cdot g_m\left(\mathbf{x}_{test}^{(i)}\right) \tag{28}$$

$$\varepsilon_{KDSR}^{(i)} = y^{(i)} - \sum_{m=1}^{M} \beta_{KDSR,m} \cdot g_m\left(\mathbf{x}_{test}^{(i)}\right) \tag{29}$$

where i=1,2,…,N_{test}. N_{test} is the size of testing data set.

After we obtain the modeling errors of the testing set, we interpolate the errors into kernel density estimation of the "golden" error distribution as demonstrated in Figure 4(b). By doing this, we get the likelihood of "observing" the modeling error, which serves as a criteria of the estimation accuracy.

As Figure 5 shows, our proposed KDSR method achieves higher likelihood. It means that the model coefficients estimated by KDSR is closer to the MLE estimation if we already know the oracle error distribution. The improvement is not quite significant, because the number of samples that are deviated from the mean value is not very large. In the following example, we will see a larger improvement of likelihood.

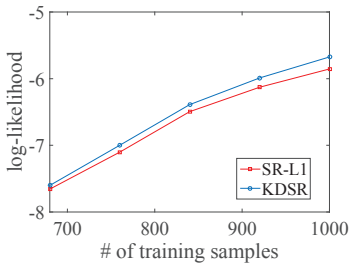
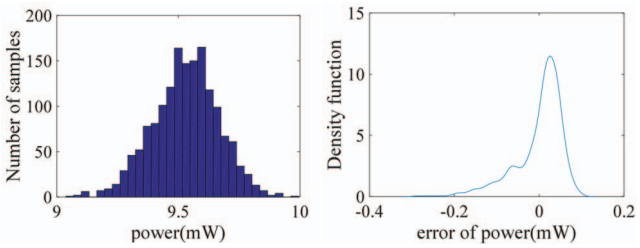


Figure 5. Error likelihood of testing set

4.2 Analog-to-Digital Converter

A flash analog-to-digital converter designed in 0.18μm CMOS process is considered in this example. There are 132 independent random variables to model both inter-die variations and intra-die mismatches. We aim to estimate power of the ADC as a linear function of these random variables. For comparison, 1604 samples are generated with schematic-level simulation.

The histogram of power is shown in Figure 6(a). The "real" distribution is obtained similarly to the op-amp example, and is shown in Figure 6(b).



(a) (b)

Figure 6. (a) The histogram of simulation data of power of ADC (b) probability density function of "golden" error

In this example, we use 800 samples from the whole data set as the testing group. We build the performance models with data points in the training group by two different approaches: SR-L1 method and KDSR method.

Figure 7 shows the modeling error likelihood for SR-L1 method and our proposed method. It is shown that given the same number of training samples, KDSR gives an estimation of model coefficients with larger likelihood. Especially, KDSR takes 220 samples to achieve larger likelihood than SR-L1 does with 260 samples. Because the simulation time of a large circuit is much longer than the execution time of the algorithm, our proposed method reduces the total time to estimate the model coefficients. In other words, to achieve the same accuracy, our proposed method achieves over 1.2x speed-up

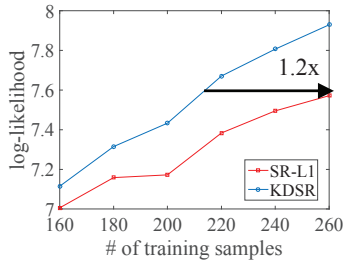


Figure 7. Error likelihood of testing set

Compared to the first example of operational amplifier, the error distribution of power has a larger ratio of samples that are deviated away from the mean value. These samples are viewed as outliers, and affect the model coefficients estimation to a great extent. As a result, KDSR shows a larger performance improvement in this ADC example, because it can detect these outliers efficiently.

Since Algorithm 1 aims to maximize the posteriori distribution as (21), we plot the value of this cost function after each iteration in EM algorithm in Figure 8. The cost function is normalized for clearance.

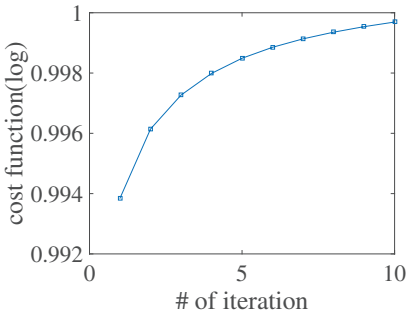


Figure 8. cost function with each iteration

As Figure 8 shows, each iteration of EM algorithm improves (21) and the cost function gradually converges. This observation is proved in [14] for kernel density based linear regression without prior knowledge.

5. CONCLUSIONS

In this paper, we proposed a KDSR method for performance modeling of AMS circuits with process variations. The proposed KDSR method uses a non-parametric kernel density estimate to approximate the likelihood function, and encodes the sparsity prior knowledge with Laplace prior distribution. An EM type algorithm is used to find the MAP estimation of model coefficients. Experimental results show that KDSR achieves higher accuracy than traditional sparse regression method. Our further work will study the kernel density based estimate with different prior information such as data collected at early-stage.

6. ACKNOWLEDGEMENTS

This work is supported partly by National Natural Science Foundation of China research project 61574046, 61474026, 61376040 and 61574044, partly by the Recruitment Program of Global Experts (the Thousand Talents Plan), partly by Shanghai Chen Guang project supported by Shanghai Municipal Education Commission and Shanghai Education Development Foundation and National Science Foundation under contract CCF-1316363 and grant 1115556.

7. REFERENCES

[1] X. Li, J. Le and L. Pileggi, *Statistical Performance Modeling and Optimization*, Now Publishers, 2007.

[2] Semiconductor Industry Associate, *International Technology Roadmap for Semiconductors*, 2011.

[3] Z. Feng and P. Li, "Performance-oriented statistical parameter reduction of parameterized systems via reduced rank regression," *IEEE ICCAD*, pp. 868-875, 2006.

[4] A. Singhee and R. Rutenbar, "Beyond low-order statistical response surfaces: latent variable regression for efficient, highly nonlinear fitting," *IEEE DAC*, pp. 256-261, 2007.

[5] T. McConaghy, "High-dimensional statistical modeling and analysis of custom integrated circuits," *IEEE CICC*, 2011.

[6] X. Li, "Finding deterministic solution from underdetermined equation: large-scale performance modeling of analog/RF circuits," *IEEE Trans. on CAD*, vol. 29, no. 11, pp. 1661-1668, Nov. 2010.

[7] Z. Wang and S. Director, "An efficient yield optimization method using a two step linear approximation of circuit performance," *IEEE EDAC*, pp. 567-571, 1994.

[8] A. Dharchoudhury and S. Kang, "Worse-case analysis and optimization of VLSI circuit performance," *IEEE Trans. on CAD*, vol. 14, no. 4, pp. 481-492, Apr. 1995.

[9] G. Debyser and G. Gielen, "Efficient analog circuit synthesis with simultaneous yield and robustness optimization," *IEEE ICCAD*, pp. 308-311, 1998.

[10] F. Schenkel, M. Pronath, S. Zizala, R. Schwencker, H. Graeb and K. Antreich, "Mismatch analysis and direct yield optimization by spec-wise linearization and feasibility-guided search," *IEEE DAC*, pp. 858-863, 2001.

[11] X. Li, P. Gopalakrishnan, Y. Xu and L. Pileggi, "Robust analog/RF circuit design with projection-based performance modeling," *IEEE Trans. on CAD*, vol. 26, no. 1, pp. 2-15, Jan. 2007.

[12] X. Li, J. Le, P. Gopalakrishnan and L. Pileggi, "Asymptotic probability extraction for nonnormal performance distributions," *IEEE Trans. on CAD*, vol. 26, no. 1, pp. 16-37, Jan. 2007.

[13] M. Sengupta, S. Saxena, L. Daldoss, G. Kramer, S. Minehane and J. Cheng, "Application-specific worst case corners using response surfaces and statistical models," *IEEE Trans. on CAD*, vol. 24, no. 9, pp. 1372-1380, 2005.

[14] W. Yao and Z. Zhao, "Kernel Density-Based Linear Regression Estimate," *Communications in Statistics-Theory and Methods*, 42(24), 4499-4512, 2013.

[15] F. Wang, W. Zhang, S. Sun, X. Li and C. Gu, "Bayesian model fusion: large-scale performance modeling of analog and mixed-signal circuits by reusing early-stage data," *IEEE DAC*, pp. 1-6, 2013.

[16] F. Wang, et al., "Co-learning Bayesian model fusion: efficient performance modeling of analog and mixed-signal circuits using side information," *IEEE ICCAD*, 2015 [in press].

[17] W. Zhang, X. Li, F. Liu, E. Acar, R. Rutenbar and R. Blanton, "Virtual probe: a statistical framework for low-cost silicon characterization of nanoscale integrated circuits," *IEEE Trans. on CAD*, vol. 30, no. 12, pp. 1814-1827, Dec. 2011.

[18] C. Bishop, *Pattern Recognition and Machine Learning*, Prentice Hall, 2007.

[19] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[20] X. Li, W. Zhang and Fa. Wang, "Large-scale statistical performance modeling of analog and mixed-signal circuits," *IEEE CICC*, 2012.

[21] W. Zhang, T. Chen, M. Ting and X. Li, "Toward efficient large-scale performance modeling of integrated circuits via multi-mode/multi-corner sparse regression," *IEEE DAC*, pp. 897-902, 2010.

[22] X. Li, "Finding deterministic solution from underdetermined equation: large-scale performance modeling by least angle regression," *IEEE DAC*, pp. 364-36, 2009.

[23] X. Li and H. Liu, "Statistical regression for efficient high-dimensional modeling of analog and mixed-signal performance variations," *IEEE DAC*, pp. 38-43, 2008.